

FAST HEURISTICS FOR MULTI-PATH SELECTION FOR MULTIPLE DESCRIPTION ENCODED VIDEO STREAMING

Ali C. Begen, Yucel Altunbasak and Ozlem Ergun

{acbegen, yucel}@ece.gatech.edu oergun@isye.gatech.edu
Georgia Institute of Technology, Atlanta, GA, USA

ABSTRACT

In a previous work [1], we proposed an optimal multi-path selection method for multiple description (MD) encoded video streaming.¹ To do so, we first modelled multi-path streaming and then developed an expression, *i.e.*, an objective (cost) function, that estimated average streaming distortion in terms of network statistics, media characteristics and application requirements. Naturally, the ultimate goal was to find the set of paths that minimized this cost function. However, finding such sets of paths turned out to be intractable in large topologies. Hence, in this paper, we provide a fast heuristics-based solution by exploiting the infrastructure features of the Internet. The simulations run over various random Internet topologies show that the proposed heuristic is able to find a *good* solution in a much shorter time than the brute-force approach. Particularly, this heuristic is best suited to such interactive multimedia applications as video-conferencing and VoIP, where multi-path computation is a time-critical process. In addition, it is also suitable for the clients whose processing power capabilities are limited.

1. INTRODUCTION

Multiple Description Coding (MDC) is a source coding that generates several encoded sub-bitstreams called *descriptions*, which are transmitted between the server and client via partially link-disjoint paths. These descriptions can still be decoded *independently*. At the client, if all descriptions are received error-free, then a maximal quality signal reconstruction is possible. However, if at least one description is available at the client, then the client can still reconstruct a signal of acceptable quality. MDC provides this robustness at the expense of some reduction in the compression efficiency. The reduced correlation between packet loss probabilities on partially link-disjoint paths makes MDC a good choice for loss resiliency. Moreover, in contrast to ARQ and FEC, MDC not only provides error resiliency features, but it also avoids excessive delays; thus, MDC has been found to be an effective solution for interactive multimedia applications in lossy networks [2, 3].

Several studies have proposed multi-path transport (path diversity) to improve the reliability of the streamed video [3, 4, 5]. Also there are some studies to model path diversity based on the loss characteristics of the paths in point-to-point and multipoint-to-point networks [6, 7]. However, these studies do not propose a formal path selection method that considers all aspects of network

statistics, media characteristics and application requirements. To address this problem, we modelled multi-path streaming and presented a formal way to evaluate the streaming distortion for each set of paths in a given network [1]. Consequently, the optimal selection was a straightforward evaluation of distortion for each possible set and choosing the one with the minimum distortion. However, the complexity of this approach increases exponentially in the total number nodes. Therefore, optimal computation may not be feasible if the client does not have enough processing power, memory space and time to find the solution or if the client resides in a network where the conditions rapidly change. To overcome these problems, we investigate fast heuristics in this paper. For the sake of continuity, we first give a brief overview of the multi-path modelling. Interested readers are referred to [1] for further details.

2. MULTI-PATH STREAMING MODELLING

Let's consider two paths, \mathcal{P}_1 and \mathcal{P}_2 , between the nodes N_S and N_C , which are referred as a *path pair*. Our ultimate goal is to find the pair that minimizes the average distortion. For this purpose, we need to estimate the end-user quality in terms of network parameters. In case of two descriptions, we can write the average distortion at the client as

$$\begin{aligned} \overline{D} = & \overbrace{\text{Prob}\{\text{Both received on time}\}}^{P_{0,0}} \times D_{0,0} \\ & + \overbrace{\text{Prob}\{1^{\text{st}} \text{ received on time \& } 2^{\text{nd}} \text{ lost or delayed}\}}^{P_{0,1}} \times D_{0,1} \\ & + \overbrace{\text{Prob}\{1^{\text{st}} \text{ lost or delayed \& } 2^{\text{nd}} \text{ received on time}\}}^{P_{1,0}} \times D_{1,0} \\ & + \overbrace{\text{Prob}\{\text{Both lost or delayed}\}}^{P_{1,1}} \times D_{1,1}. \end{aligned} \quad (1)$$

Here, $D_{m,n}$ and $P_{m,n}$ ($m, n \in \{0, 1\}$) denote a specific distortion and its occurring probability, respectively as defined in Table 1. In order to capture the characteristics of multi-path streaming, we seek a simple yet representative model. Fig. 1 depicts our end-to-end multi-path model. In this figure, we have two paths between the server and client that split at node N_Z . We combine the joint links between the server and N_Z to form a sub-network denoted by $\text{Network}_{\text{joint}}$. On $\text{Network}_{\text{joint}}$, the two correlated descriptions are supposed to be transmitted back-to-back and hence, they are in the same burst period. On the other hand, we consider the rest of the paths as disjoint despite a possible merge at a later node. That is, the two correlated descriptions do not fall in the same burst period implying that their loss events beyond N_Z will be independent

This work is supported by NSF under NSF CAREER award CCR-0133221.

¹It should be mentioned that the proposed multi-path selection method is *optimal* within the context of the network models and objective functions that are considered in [1].

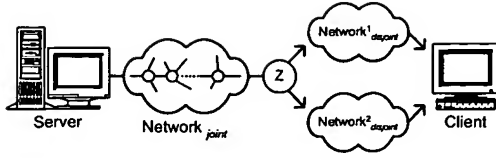


Fig. 1. End-to-end multi-path model.

of each other. This is a valid approximation as long as the disjoint links have a delay difference larger than the average burst period between them [1]. This is typically true because of the unevenly experienced delays and jitters on disjoint links. Thus, the links beyond N_Z are combined to form the respective $\text{Network}_{disjoint}^i$. Note that node N_Z might be the server, which implies that the paths are totally link-disjoint, i.e., Network_{joint} is an empty set. The pertaining parameters to our model are tabulated in Table 1. We start our derivations with the distortion.

In compressed-video applications, the distribution of Discrete Cosine Transform (DCT) coefficients can be modelled as Laplacian. For a Laplacian source with variance σ^2 , the distortion as a function of the aggregated bandwidth of the path pair (B) can be approximated by [8]

$$D(B) \sim \sigma^2 2^{-2 \times \frac{B}{W \times H \times F \times c}}, \quad (2)$$

where $W \times H$ (pixels/frame) and F (frames/second) represent the frame resolution and frame rate, respectively. c is a known constant depending on the chroma subsampling format. Given this relation, we can compute the distortion terms in (1) by plugging the corresponding B values into (2). First, for $D_{0,0}$ the aggregated bandwidth is typically the sum of the individual bandwidths, i.e., $B = B_1 + B_2$ provided that the joint links have available bandwidths higher than this sum. However, if this is not the case, B is then limited by the bottleneck joint-link capacity. Second, to compute $D_{0,1}$ and $D_{1,0}$, we take $B = B_1$ and $B = B_2$, respectively. Finally, $D_{1,1}$ is computed by letting $B = 0$.

The next step in the computation of (1) is to find $P_{0,0} - P_{1,1}$, each of which is comprised of two components. The first component, *arrival probability*, computes the probability that the description is not lost during the transmission and the second component, *on-time delivery probability*, computes the probability that the description arrival occurs before a pre-specified deadline.

We adopted a well-known two-state Markovian Gilbert-Elliott (GE) model to describe the temporal behavior of packet losses on a link. We give the transition probabilities, α and β , for the link $l_{u,v}$ by

$$\alpha_{u,v} = \frac{p_{u,v} \times \beta_{u,v}}{1 - p_{u,v}} \quad \text{and} \quad \beta_{u,v} = \frac{1}{L_{u,v}}. \quad (3)$$

We employ a joint GE model for Network_{joint} and individual Bernoulli models for $\text{Network}_{disjoint}^i$ based on our model. To find the joint GE parameters, namely α_{joint} and β_{joint} , we first approximate the joint packet loss rate and average burst length for Network_{joint} as follows [1]:

$$p_j = 1 - \prod_{\forall l_{u,v} \in \text{Network}_{joint}} (1 - p_{u,v}) \quad (4)$$

$$L_j = \frac{1}{p_j} \times \sum_{\forall l_{u,v} \in \text{Network}_{joint}} (p_{u,v} \times L_{u,v}) \quad (5)$$

Table 1. Notation and model parameters.

N_k	Node k
$l_{u,v}$	Link between the nodes N_u and N_v
$b_{u,v}$	Bandwidth on link $l_{u,v}$
$p_{u,v}$	Packet loss rate on link $l_{u,v}$
$t_{u,v}$	Minimum (constant portion) delay on link $l_{u,v}$
$j_{u,v}$	Jitter on link $l_{u,v}$
\mathcal{P}_i	Path i
$D_{m,n}$	Distortion. m and n denote the arrivals of 1 st and 2 nd descriptions, respectively.
	$0 \rightarrow$ arrival on time. $1 \rightarrow$ loss or excessively delayed.
$P_{m,n}$	Probability corresponding to $D_{m,n}$
B_i	End-to-end bandwidth on path \mathcal{P}_i
π_A	Steady-state prob. for state A (arrival). $\pi_A = \frac{\beta}{\alpha + \beta}$
π_L	Steady-state prob. for state L (loss). $\pi_L = \frac{\alpha}{\alpha + \beta}$
α	Transition prob. from A to L
β	Transition prob. from L to A
$L_{u,v}$	Average burst length on link $l_{u,v}$
p_j	Packet loss rate on Network_{joint}
L_j	Average burst length on Network_{joint}
p_d^i	Packet loss rate on $\text{Network}_{disjoint}^i$
j_j	Jitter on Network_{joint} .
	$j_j = \sum_{\forall l_{u,v} \in \text{Network}_{joint}} j_{u,v}$
j_d^i	Jitter on $\text{Network}_{disjoint}^i$.
	$j_d^i = \sum_{\forall l_{u,v} \in \text{Network}_{disjoint}^i} j_{u,v}$
J_i	End-to-end jitter on path \mathcal{P}_i . $J_i = j_j + j_d^i$

Then, we compute α_{joint} and β_{joint} by plugging p_j and L_j into (3). On the other hand, since they are modelled by Bernoulli, the packet loss rates in $\text{Network}_{disjoint}^i$ are given by

$$p_d^i = 1 - \prod_{\forall l_{u,v} \in \text{Network}_{disjoint}^i} (1 - p_{u,v}). \quad (6)$$

The second step in computing the probability $P_{m,n}$ is to derive the on-time delivery probabilities. Usually, different paths have different tolerances for jitter. We define t_{max}^i as the maximum jitter that can be tolerated for path \mathcal{P}_i . Let $T_i = \sum_{\forall l_{u,v} \in \mathcal{P}_i} t_{u,v}$ represent the end-to-end minimum delay on path \mathcal{P}_i . Then, we have $t_{max}^i = T_{deadline} - T_i$ where $T_{deadline}$ is the tolerable delay for the target application. Now, the on-time delivery probability can also be given as the probability that J_i is smaller than t_{max}^i .

Finally, we have all the information required to write success probabilities ($P_{0,0} - P_{1,1}$) explicitly. The corresponding equations are given in (7) - (10) where we denote the joint GE parameters by α and β instead of α_{joint} and β_{joint} , respectively.

$$P_{0,0} = ((\pi_A(1 - \alpha)^2 + \pi_L\beta(1 - \alpha)) \times (1 - p_d^1) \times (1 - p_d^2)) \times \text{Prob}(j_j + j_d^1 \leq t_{max}^1 \text{ and } j_j + j_d^2 \leq t_{max}^2) \quad (7)$$

$$P_{0,1} = ((\pi_A(1 - \alpha)\alpha + \pi_L\beta\alpha) \times (1 - p_d^1)) \times \text{Prob}(j_j + j_d^1 \leq t_{max}^1) + ((\pi_A(1 - \alpha)^2 + \pi_L\beta(1 - \alpha)) \times (1 - p_d^1) \times p_d^2) \times \text{Prob}(j_j + j_d^1 \leq t_{max}^1) + ((\pi_A(1 - \alpha)^2 + \pi_L\beta(1 - \alpha)) \times (1 - p_d^1) \times (1 - p_d^2)) \times \text{Prob}(j_j + j_d^1 \leq t_{max}^1 \text{ and } j_j + j_d^2 > t_{max}^2) \quad (8)$$

$$P_{1,0} = \left((\pi_A \alpha \beta + \pi_L (1 - \beta) \beta) \times (1 - p_d^2) \right) \quad (9)$$

$$\begin{aligned} & \times \text{Prob}(j_j + j_d^2 \leq t_{max}^2) \\ & + \left((\pi_A (1 - \alpha)^2 + \pi_L \beta (1 - \alpha)) \times p_d^1 \times (1 - p_d^2) \right) \\ & \times \text{Prob}(j_j + j_d^2 \leq t_{max}^2) \\ & + \left((\pi_A (1 - \alpha)^2 + \pi_L \beta (1 - \alpha)) \times (1 - p_d^1) \times (1 - p_d^2) \right) \\ & \times \text{Prob}(j_j + j_d^1 > t_{max}^1 \text{ and } j_j + j_d^2 \leq t_{max}^2) \end{aligned}$$

$$P_{1,1} = 1 - P_{0,0} - P_{0,1} - P_{1,0} \quad (10)$$

3. SOLUTION APPROACHES

Once we have defined the cost function, we seek the optimal path pair such that the expression in (1) is minimized. Given that (1) is highly nonlinear, solving this optimization problem may become computationally intractable in large topologies. In this section, we study the brute-force complete enumeration and a heuristic that selects a good pair of diverse paths. This heuristic is based on first identifying a small subset of good paths, and then choosing the pair that minimizes (1) among this subset. In developing our heuristic, we exploit the routing hierarchy in the Internet. See [9] for further details on this subject.

3.1. Brute-Force Approach

Consider an Internet topology with M transit domains (TD) each of which contains N nodes. An illustrative topology with four TDs is shown in Fig. 2. Assume that multi-path routing capability is enabled only within TDs, but not stub domains (SD). In this topology, the brute-force (BF) approach for finding the optimal pair enumerates all possible paths between the server and client. Then, (1) is computed for each pair and the pair that gives the minimum average distortion is selected.

The complexity analysis of this approach can be summarized as follows: All paths in the given topology can be generated in $O(2^{MN})$ time and (1) is evaluated for all possible pairs in $O(MN \times 2^{2MN})$ time. Finally, identification of the optimal pair requires $O(\log 2^{2MN})$ time. Hence, the BF approach has a complexity of

$$O(2^{MN}) + O(MN \times 2^{2MN}) + O(MN) = O(MN \times 2^{2MN}). \quad (11)$$

In (11), we observe that the BF approach complexity increases exponentially with both M and N . Below, we develop a more efficient heuristic that is referred as *rapid path generation* (RPG).

3.2. Rapid Path Generation

An important characteristic of the Internet is that a path connecting two nodes in different SDs, e.g., the ones containing the server and client, will never pass through any other SDs [9]. This feature implies that multi-path routing will mainly take place within or between the TDs. Moreover, between two TDs, usually there is only one link through which all traffic from the source domain is carried to the destination domain. This natural decomposition of the Internet leads us to consider the TDs individually in the path generation process.

We start our algorithm by finding good paths in each TD. During this process, we also exploit the fact that among the links in a TD, only a few of them may be heavily congested at a time while the rest is lightly congested or not congested at all. Eliminating these congested links clearly reduces the topology size. Moreover, after this reduction the remaining links will have similar (almost

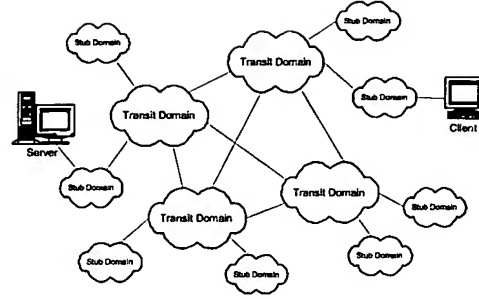


Fig. 2. An illustrative Internet topology with four TDs.

zero) packet loss rates. On the other hand, the end-to-end bandwidth is often limited by the SD or inter-domain links. We suppose that TD links can support the bandwidth desired by the clients. Hence, the remaining TD links' bandwidths have negligible effect in selecting the good paths.

Now, we are left only with the delay and jitter parameters in the path generation process. To find a subset of good paths in each TD, we solve a k -shortest path algorithm where the cost on each link is the sum of its delay and average jitter. The most straightforward k -shortest path algorithm for M TDs requires $O(kMN^2)$ time. After selecting k good paths in each TD, we combine them to create $O(k^M)$ end-to-end paths. Then, out of these paths we form $O(k^{2M})$ pairs. At this point, we evaluate (1) for each pair. This process takes $O(MN \times k^{2M})$ time. Finally, selecting the pair with minimum average distortion can be accomplished in $O(\log k^{2M})$ time. Hence, the RPG approach has an overall complexity of

$$O(kMN^2) + O(MN \times k^{2M}) + O(M) = O(MN \times k^{2M}). \quad (12)$$

Generally, we have $M < N$, which is the case in small-to-moderate sized networks. Hence, although the RPG approach runs in exponential time in M , it still provides an important improvement over the BF complete enumeration, which requires exponential time in the total number of nodes ($M \times N$). In addition, with this approach rather than trying to seek the solution in the whole network, we handle reduced-sized problems in each TD separately. Hence, the demand for the memory space in the RPG implementation is substantially lower than that in the BF implementation. On the other hand, k is an important parameter in (12). We will see its impact on the performance of RPG approach in the next section.

3.3. Performance Comparisons

In order to evaluate the performance of RPG, we used random Internet topologies generated by GT-ITM [9]. We varied k , number of TDs and SDs to see the effects of variations in network connectivity. The link parameters were assigned random values as described in [1]. For each random topology, we first found the path pairs corresponding to the BF and RPG solutions. Then, by using each pair we streamed a standard test sequence from the server to the client with a delay tolerance of 200 ms.² We repeated this process 100 times to obtain reliable peak signal-to-noise ratio (PSNR) values. The results are tabulated in Tables 2 - 4 along with the values of k , M and N . The PSNR differences between the BF and

²We used TABLE TENNIS sequence. This sequence comprised of 150 frames each of which was 352 x 240 pixels. The frame rate was 30 f/s.

RPG approaches are also given in the last column of each table. Note that each row represents a different random topology. Hence, the results in a row of a table should only be compared with the results in the same row of other tables.

Table 2. Comparison of the BF and RPG approaches when $k = 1$.

k	M	N	BF	RPG	Difference
1	2	5	33.39 dB	31.82 dB	1.57 dB
1	2	8	33.45 dB	31.56 dB	1.89 dB
1	3	5	33.36 dB	31.63 dB	1.73 dB
1	3	8	33.38 dB	31.44 dB	1.94 dB
1	4	8	33.30 dB	31.39 dB	1.91 dB

Table 3. Comparison of the BF and RPG approaches when $k = 2$.

k	M	N	BF	RPG	Difference
2	2	5	33.39 dB	32.93 dB	0.46 dB
2	2	8	33.45 dB	32.85 dB	0.60 dB
2	3	5	33.36 dB	32.84 dB	0.52 dB
2	3	8	33.38 dB	32.66 dB	0.72 dB
2	4	8	33.30 dB	32.43 dB	0.87 dB

Table 4. Comparison of the BF and RPG approaches when $k = 3$.

k	M	N	BF	RPG	Difference
3	2	5	33.39 dB	33.32 dB	0.07 dB
3	2	8	33.45 dB	33.30 dB	0.15 dB
3	3	5	33.36 dB	33.24 dB	0.12 dB
3	3	8	33.38 dB	33.15 dB	0.23 dB
3	4	8	33.30 dB	33.06 dB	0.24 dB

The results show that as we identify more good paths in each TD, the performance of RPG approach becomes closer to the optimal in all topologies. Although there is about 1.80 dB difference in the average qualities when $k = 1$, this difference reduces to about 0.16 dB when $k = 3$. This is also shown in Fig. 3 and 4 where individual frame qualities are plotted. In Fig. 4 the RPG plot tracks the BF plot more closely on the average than it does in Fig. 3. Even when $k = 3$ (and $M = 4$, $N = 8$), the number of operations required by the RPG approach is 500K times smaller than the one required by the BF approach, which proves a quite large savings in terms of processing power.

Another interesting point is the choice of k . In addition to the reported results, we also tested the same topologies with $k = 4$. However, the improvement over the case with $k = 3$ was not that significant. Hence, we conclude that keeping $k = 3$ is a good choice. It is worthy to note that $k = 1$ does not necessarily imply that the 1st and 2nd shortest paths are selected between the server and client. Although only the shortest path is computed in each TD for $k = 1$, we still evaluate (1) for all possible end-to-end pairs and choose the pair accordingly.

4. CONCLUSION

In this paper, we presented a fast heuristics-based solution for the optimal multi-path selection problem introduced in [1]. This approach performs within 0.2 dB of the optimal solution in small-to-moderate sized networks without incurring any run-time or memory space problems. Particularly, this fast heuristic is best suited to such interactive multimedia applications as video-conferencing and VoIP, where multi-path computation is a time-critical process. In addition, it is also suitable for the clients whose processing power capabilities are limited.

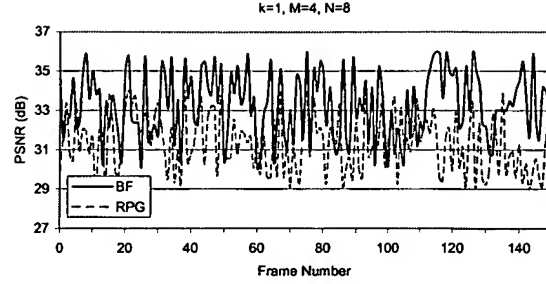


Fig. 3. The PSNR comparisons of the BF and RPG approaches when $k = 1$, $M = 4$ and $N = 8$.

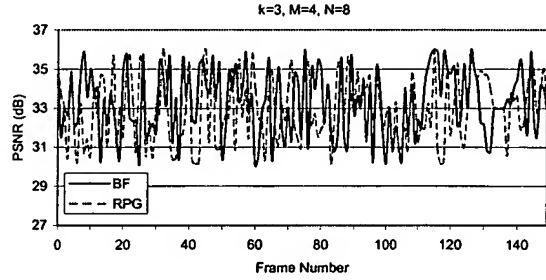


Fig. 4. The PSNR comparisons of the BF and RPG approaches when $k = 3$, $M = 4$ and $N = 8$.

5. REFERENCES

- [1] Ali C. Begen, Yucel Altunbasak, and Ozlem Ergun, "Multi-path selection for multiple description encoded video streaming," in *IEEE ICC*, 2003.
- [2] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Processing Mag.*, vol. 18, no. 5, pp. 74–93, 2001.
- [3] J. G. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in *VCIP*, 2001.
- [4] S. Lin, S. Mao, Y. Wang, and S. Panwar, "A reference picture selection scheme for video transmission over ad-hoc networks using multiple paths," in *IEEE ICME*, 2001.
- [5] R. Chow, C. Lee, and Jonathan C.L. Liu, "Traffic dispersion strategies for multimedia streaming," in *IEEE Eighth Workshop on Future Trends of Distributed Computing Systems*, 2001.
- [6] J.G. Apostolopoulos, W. Tan, S.J. Wee, and G.W. Wornell, "Modeling path diversity for multiple description video communication," in *IEEE ICASSP*, 2002.
- [7] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee, "On multiple description streaming with content delivery networks," in *IEEE INFOCOM*, 2002.
- [8] John G. Proakis, *Digital Communications*, McGraw-Hill, fourth edition, 2000.
- [9] Ellen W. Zegura, Kenneth L. Calvert, and Michael J. Donahoo, "A quantitative comparison of graph-based models for Internet topology," *IEEE/ACM Trans. Networking*, vol. 5, no. 6, pp. 770–783, 1997.